tutorial

Il protocollo Modbus

Un protocollo di comunicazione di facile implementazione e adatto a essere integrato nelle reti Ethernet

Massimo Giussani

Modbus è un protocollo di comunicazione di alto livello basato sullo scambio di messaggi tra dispositivi in modalità master-slave e client-server, caratterizzato dalle specifiche aperte e da un'implementazione particolarmente semplice.
Inizialmente sviluppato da Modicon (gruppo Schneider) per il trasferimento di dati di controllo tramite interfacce seriali RS-232, il protocollo ha conosciuto una seconda primavera con l'introduzione di una variante con incapsulamento TCP-IP e la cessione delle specifiche da parte di Schneider a un'organizzazione no-profit (Modbus-IDA, www.modbus.org). Modbus trova applicazione nel controllo e nella configurazione delle apparecchiature più disparate, e in particolare in ambito industriale per le comunicazioni di sensori e attuatori con controllori, interfacce uomo-macchina (HMI) e PC di supervisione.

Le varianti principali

Modbus è presente nel panorama industriale in numerose forme che si possono tuttavia ricondurre a tre varianti fondamentali. La più antica (che può essere identificata con Modbus/Ascii e Modbus/RTU) permette di stabilire comunicazioni seriali asincrone su interfacce RS-232 e RS-485, ed è stata adattata anche a mezzi di trasmissioni diversi dal rame, come la fibra ottica e i collegamenti radio. La seconda variante, che sta conoscendo un momento di particolare fulgore, si appoggia a uno stack TCP-IP per consentire la

Confronto tra Modbus/Ascii and Modbus/RTU

	Modbus/Ascii	**Modbus/RTU**
Caratteri usati	Simboli Ascii delle cifre esadecimali 0, 1, 2, ... E, F	Valori binari compresi tra 0 e 255
Inizio frame	Carattere ':' la durata di un byte	Silenzio di 3,5 volte
Fine frame	Sequenza CR/LF la durata di un byte	Silenzio di 3,5 volte
Bit iniziale	1	1
Bit dati	7	8
Pausa nel messaggio	1 sec	1,5 volte la durata di un byte
Controllo di ridondanza	LRC Longitudinal Redundancy Check	CRC Cyclic Redundancy Check

Tab. 1 - Differenze essenziali tra le modalità Ascii e RTU

comunicazione su reti Ethernet. E' nota con i nomi di Modbus/TCP, Modbus/IP e Modbus/Ethernet. Esiste infine una terza versione estesa e mirata alle reti ad alte prestazioni basate sul passaggio di token: si tratta di una variante proprietaria denominata Modbus Plus (spesso indicata con MB+). Sebbene Modbus sia ancora un marchio registrato di proprietà di Schneider Automation, non ci sono licenze aggiuntive da pagare per il solo impiego dei protocolli Modbus e Modbus TCP/IP. La versione estesa Modbus Plus conserva ancora il carattere di protocollo proprietario.

Semplice comunicazione seriale

Nella sua forma originale, Modbus consente il funzionamento in modalità half-duplex e full-duplex su reti seriali RS-485 e RS-232, con i bit dati rappresentati da tensioni positive e negative sui terminali Rx e Tx della porta seriale. L'interazione tra i dispositivi è di tipo master-slave e sono previsti due tipi di trasmissione: Ascii e RTU. Nella modalità Ascii i singoli byte da otto bit che costituiscono il messaggio sono trasmessi sotto forma di due caratteri Ascii che rappresentano la codifica esadecimale del valore; durante il funzionamento in modalità RTU (Remote Terminal Unit) viene inviata la rappresentazione binaria diretta del valore. E' evidente che la modalità RTU, a parità di velocità di trasmissione, permette di trasmettere una maggior quantità di informazioni. Un punto a favore della modalità Ascii sta nel fatto che sono ammesse pause fino a un secondo tra un carattere e l'altro senza che questo provochi un errore di comunicazione. La scelta del tipo di codifica dati da utilizzare viene solitamente effettuata dall'utilizzatore nella fase di configurazione dei nodi della rete, insieme all'impostazione dei parametri di comunicazione della porta seriale. La tabella 1 mostra le principali differenze tra i due dialetti. Uno dei principali problemi di Modbus è che quando ci sono decine o centinaia di nodi connessi in multidrop a una stessa rete il polling dei dispositivi nella modalità di comunicazione del tipo master-slave può comportare un considerevole rallentamento delle comunicazioni.

Le unità dati PDU e ADU

Una tipica comunicazione via Modbus consiste essenzialmente di tre stadi: la formulazione di una richiesta da parte di un dispositivo a un altro, l'esecuzione delle azioni necessarie a soddisfare la richiesta, e la restituzione al dispositivo iniziale delle informazioni risultanti, siano esse l'effettivo risultato dell'elaborazione o un codice di errore derivante dall'impossibilità di portare a termine il compito. Le informazioni vengono scambiate sotto forma di unità dati indipendenti dai livelli sottostanti nella pila Iso-Osi. Elemento fondamentale nello scambio dati in Modbus è la Protocol Data Unit (PDU) costituita da un campo che contiene il codice funzione (codificato con un solo byte) e un campo dati di lunghezza variabile, eventualmente nulla, che contiene il corpo del messaggio. L'integrazione all'interno di altre reti può richiedere dei campi aggiuntivi che sono raccolti nella Application Data Unit (ADU). Generalmente è presente un campo indirizzi che permette di identificare il dispositivo che ha effettuato la richiesta, e un campo con i codici per la correzione degli errori. La figura 1 illustra il tipico frame Modbus nelle sue due varianti. Il protocollo prevede tre diversi tipi di PDU: richiesta (mb_req_pdu), risposta (mb_rsp_pdu) e risposta con eccezione (mb_excep_rsp_pdu). Un dispositivo client inoltra al server la propria richiesta di eseguire una determinata azione, sostanziandola con informazioni aggiuntive nel campo dati (un esempio potrebbe essere la richiesta di leggere la temperatura di un particolare sensore connesso alla sottorete gestita dal dispositivo server). Il dispositivo server, ricevuta la richiesta e verificata la validità del codice funzione, esegue l'operazione richiesta. Se l'operazione ha esito positivo, viene generato un frame contenente le informazioni risultanti (il valore di temperatura opportunamente codificato e memorizzato in un registro, ad esempio) e il relativo codice funzione che viene poi immesso sul bus; se la verifica dei dati o l'operazione non sono state portate a termine o hanno generato errori, nel campo del codice funzione viene restituito un codice di errore (pari al codice funzione della chiamata incrementato di 127), mentre nel campo dati viene riportato il codice dell'eccezione che ha determinato l'impossibilità a rispondere.

Frame Modbus/Ascii

Inizio	Indirizzo	Codice funzione	Campo dati	LRC check	Fine
1 carattere	2 caratteri	2 caratteri	N caratteri	2 caratteri	CR+LF

Frame Modbus/RTU

Inizio	Indirizzo	Codice funzione	Campo dati	LRC check	Fine
Silenzio di durata 3,5 byte	1 byte	1 byte	N byte	2 byte	Silenzio di durata 3,5 byte

Fig. 1 - Formato dei frame Modbus nelle varianti Ascii e RTU

I campi indirizzo e funzione

Quando un dispositivo master richiede dei dati, invia come primo byte un codice che rappresenta l'indirizzo della periferica slave da interrogare. Ogni periferica è dotata di un indirizzo univoco rappresentato da un numero compreso tra 1 e 247 ed è così in grado di sapere se continuare a legge-

re il messaggio o ignorarlo subito dopo aver letto il primo byte. Molte varianti del protocollo consentono di utilizzare due byte per l'indirizzo di periferica, portando il numero di nodi a un meno restrittivo limite di 65.635. La seconda informazione trasmessa è il codice funzione, un valore compreso tra 1 e 255, che specifica il tipo di azione che deve essere (o è stata) eseguita. I codici si possono riferire a funzioni già incluse nello standard (funzioni pubbliche) o a estensioni programmate dall'utente. Le funzioni pubbliche sono state convalidate dalla comunità di sviluppatori e utilizzatori Modbus, sono documentate nella RFC Ietf relativa allo standard Modbus e rappresentano una base condivisa da tutti i dispositivi che si conformano ad esso. Alcuni produttori possono decidere di implementare funzioni particolari utilizzando uno dei codici riservati agli utenti, cosa che può essere fatta senza dover richiedere l'approvazione dell'ente di riferimento: è tuttavia bene tenere presente che questa libertà espone il prodotto a potenziali incompatibilità con applicazioni di terze parti che potrebbero utilizzare il medesimo codice per altri compiti.

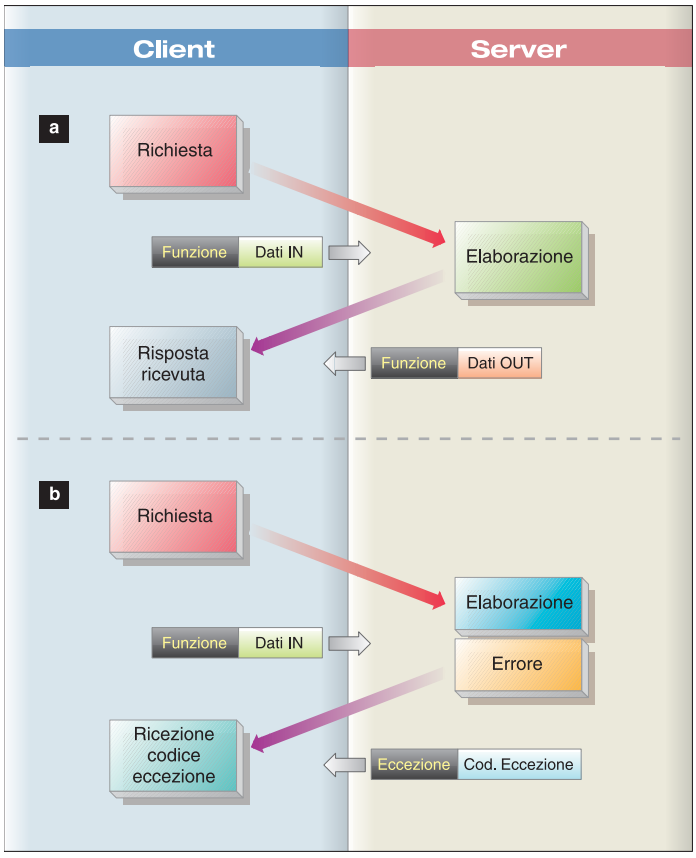

Fig. 2 - Due tipiche transazioni Modbus: a) Richiesta di informazioni con risposta; b) Generazione di un'eccezione

Il campo dati

Il campo dati passato dal client al server nella fase di richiesta può essere vuoto (nel caso in cui non siano necessarie ulteriori informazioni per portare a termine il compito indicato dal codice funzione) o può contenere una serie di informazioni che dettagliano il tipo operazione da svolgere con un eventuale codice di sottofunzione, le variabili coinvolte, i registri da utilizzare, eventuali dati da trasferire nel dispositivo ricevente e via di seguito. Analogamente il campo dati restituito dal server riporterà tutte le informazioni pertinenti al compito svolto, con il tipo e numero di variabili restituite. La lunghezza del campo dati è limitata a un valore ereditato dalle prime versioni del protocollo. La versione di Modbus su linea seriale può contare su una dimensione massima del frame ADU di 256 byte; di questi, un byte è utilizzato per l'indirizzo del server, due per il controllo ciclico di ridondanza (CRC) e i restanti 253 vanno a costituire l'unità dati del protocollo. Tolto un byte per il codice funzione/eccezione, ai dati rimangono 252 byte da occupare. La versione per TCP-IP richiede 7 byte aggiuntivi alla PDU, e finisce per occupare un totale di 260 byte. La codifica utilizzata per la trasmissione di numeri che richiedono più byte è di tipo big-endian.

I dispositivi memorizzano i dati scambiati in quattro tabelle da 9.999 valori ciascuna. Due tabelle riportano valori discreti (lo stato acceso-spento di un relè), e altre due i valori analogici (registri) in parole di 16 bit, la cui interpre-

Codice funzione	Azione	Effettuata sulla tabella
01 (01h)	Read	Discrete Output Coils
02 (02h)	Read	Discrete Input Contacts
03 (03h)	Read	Analog Output Holding Registers
04 (04h)	Read	Analog Input Registers
05 (05h)	Write single	Discrete Output Coils
15 (0Fh)	Write multiple	Discrete Output Coils
06 (06h)	Write single	Analog Output Holding Registers
16 (10h)	Write multiple	Analog Output Holding Registers

Tab. 2 - Principali codici di funzione pubblici di un comando Modbus

tazione dipende dalla particolare implementazione. I valori sono letti o scritti facendo riferimento agli indirizzi dati delle rispettive tabelle, ossia valori esadecimali compresi tra 0000h e 270Eh, al netto del relativo offset. E' possibile estendere il numero di registri analogici in uscita utilizzando anche gli indirizzi compresi tra 270Fh e FFFFh, ma è necessario assicurarsi che i dispositivi utilizzati supportino il range esteso. L'indirizzamento è comunque limitato a 16 bit, cosa che comporta un limite massimo per i valori per gli indirizzi pari a 65.535. Ogni produttore di dispositivi Modbus deve dotare il proprio prodotto di un mezzo per tradurre (rimappare) i valori impiegati nelle effettive locazioni di memoria utilizzate per memorizzare i dati di funzionamento. In questo modo si crea un livello di astrazione che permette agli utilizzatori Modbus di accedere ai parametri di funzionamento del dispositivo in maniera trasparente e indipendente dai dettagli della particolare implementazione.

Indirizzi dati	Offset	Numero consecutivo associato all'elemento analogico o discreto	Tipo	Nome tabella
0000h - 270Eh	00001	00001-09999	R/W	Discrete Output Coils
0000h - 270Eh	10001	10001-19999	Read-Only	Discrete Input Contacts
0000h - 270Eh	30001	30001-39999	Read-Only	Analog Input Registers
0000h - 270Eh	40001	40001-49999	R/W	Analog Output Holding Registers

Tab. 3 - Allocazione dei dati discreti e analogici in un dispositivo Modbus

Un codice di ridondanza ciclica (CRC/LRC check), per la verifica dell'integrità dei dati ricevuti, chiude il messaggio. I frame che non soddisfano il controllo possono così venire rigettati. La figura 3 mostra un esempio di comunicazione Modbus in cui un dispositivo master invia sul bus un comando per la lettura del valore analogico contenuto nel registro 30.009 del dispositivo con indirizzo 17. Il dispositivo interpreta il messaggio, legge il valore del corrispondente registro e lo immette sul bus.

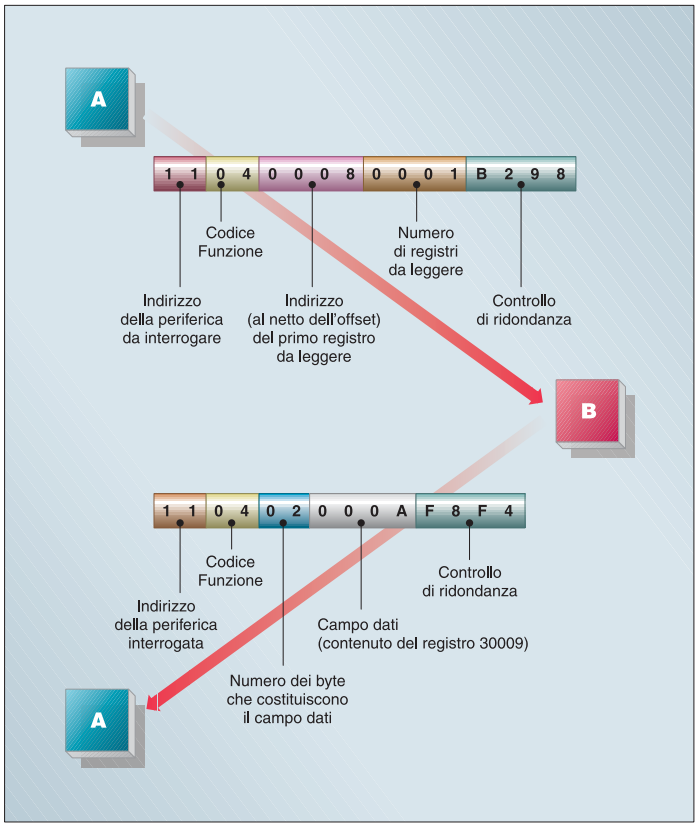

Fig. 3 - Esempio di comunicazione Modbus

Modbus/TCP

La variante TCP del protocollo è sostanzialmente identica alla versione seriale originale alla quale viene aggiunta un modulo per l'incapsulamento TCP/IP. Questo rende di fatto il protocollo un vero e proprio protocollo Internet (IP) e apre la strada all'impiego anche sulle comuni reti di comunicazione da ufficio. Un qualunque computer connesso in rete può agire da client o server Modbus scambiando messaggi tramite la porta riservata 502 dello stack TCP-IP. Il principale vantaggio di questo approccio sta nella modalità di interazione tra i vari nodi della rete: essendo di tipo client-server, ogni dispositivo server è in grado di scambiare dati in maniera simultanea con più di-spositivi client. La comunicazione tra la rete Ethernet e i differenti mondi seriali (RS-232, RS-485, wireless, e altro ancora) richiede l'impiego di un dispositivo di gateway che effettui la traduzione da e verso la variante TCP (aggiungendo e rimovendo l'incapsulamento dei dati). In quest'ottica Modbus si presta alla realizzazione di reti eterogenee che integrano dispositivi differenti per tipologia e modalità di funzionamento. Ovviamente le prestazioni dei trasferimenti dati attraverso una normale rete Internet non consentono la realizzazione di sistemi deterministici, ma in questa incarnazione il protocollo Modbus può essere proficuamente utilizzato ai fini di supervisione, manutenzione preventiva e, più in generale, per la comunicazione a distanza con dispositivi intelligenti (configurazione e diagnostica).

Una rete Intranet dedicata e ad alte prestazioni, basata su un'infrastruttura Ethernet commutata ad alta velocità, può inoltre permettere la realizzazione di reti di controllo con un livello di determinismo accettabile per numerose applicazioni di controllo e automazione industriale. ■